

**AMENDMENTS TO THE CLAIMS**

*Please enter the following amendments:*

1. (Currently Amended) A method of processing data in a programmable processor, the method comprising:

decoding a single instruction for selectively arranging data, specifying a data selection operand and a first and a second register each having a register width, the first and second registers providing a plurality of data elements each having an elemental width smaller than the register width, the data selection operand comprising a plurality of fields each independently selecting one of the plurality of data elements; and

for each field of the data selection operand, providing the data element selected by the field to a predetermined position in a catenated result.

2. (Previously Presented) The method of claim 1 wherein each field of the data selection operand provides a sufficient number of bits to specify any one of the plurality of data elements.

3. (Original) The method of claim 2 wherein each field of the data selection operand has a width of  $n$  bits, wherein the plurality of data elements comprises  $2^n$  data elements.

4. (Original) The method of claim 1 wherein the data selection operand is provided by a register specified by the single instruction.

5. (Original) The method of claim 4 wherein the data selection operand has a width equal to the specified register width.
6. (Original) The method of claim 1 wherein the catenated result is provided to a register.
7. (Original) The method of claim 1 wherein the plurality of data elements has a combined width equal to the width of the first register plus the width of the second register.
8. (Original) The method of claim 1 wherein the instruction further specifies a data element width of the plurality of data elements.
9. (Original) The method of claim 1 wherein each data element has a width of 8 bits.
10. (Original) The method of claim 1 wherein the catenated result has a width of 128 bits.
11. (Original) The method of claim 1 wherein for each field of the data selection operand, a relative location of the field within the data selection operand corresponds to a relative location of the predetermined position within the catenated result.

12. (Previously Presented) The method of claim 1 further comprising:  
decoding a second single instruction specifying a third and a fourth register each containing a plurality of floating-point operands;  
multiplying the plurality of floating point operands in the third register by the plurality of operands in the fourth register to produce a plurality of products; and  
providing the plurality of products to partitioned fields of a result register as a catenated result.

13. (Currently Amended) A method for selectively arranging data in a programmable processor, the method comprising:  
decoding a single instruction specifying a data selection operand and a first register having a register width, the first register providing a plurality of data elements each having an elemental width smaller than the register width, the data selection operand comprising a plurality of fields each independently selecting one of the plurality of data elements; and  
for each field of the data selection operand, providing the data element selected by the field to a predetermined position in a catenated result.

14. (Currently Amended) A computer-readable medium:  
having instructions that instruct a computer system to perform operations,  
at least some of the instructions including a group element selection instruction for  
selectively arranging data in a programmable processor, the group element selection instruction  
capable of instructing a computer to perform operations comprising:

decoding the group element selection instruction specifying a data selection  
operand and a first and a second register each having a register width, the first and second  
registers providing a plurality of data elements each having an elemental width smaller than the  
register width, the data selection operand comprising a plurality of fields each independently  
selecting one of the plurality of data elements; and

for each field of the data selection operand, providing the data element selected by  
the field to a predetermined position in a catenated result.

15. (Previously Presented) The computer-readable medium of claim 14 wherein each  
field of the data selection operand provides a sufficient number of bits to specify any one of the  
plurality of data elements.

16. (Original) The computer-readable medium of claim 15 wherein each field of the data  
selection operand has a width of  $n$  bits, wherein the plurality of data elements comprises  $2^n$  data  
elements.

17. (Original) The computer-readable medium of claim 14 wherein the data selection  
operand is provided by a register specified by the single instruction.

18. (Original) The computer-readable medium of claim 17 wherein the data selection operand has a width equal to the specified register width.

19. (Original) The computer-readable medium of claim 14 wherein the catenated result is provided to a register.

20. (Original) The computer-readable medium of claim 14 wherein the plurality of data elements has a combined width equal to the width of the first register plus the width of the second register.

21. (Original) The computer-readable medium of claim 14 wherein the instruction further specifies a data element width of the plurality of data elements.

22. (Original) The computer-readable medium of claim 14 wherein each data element has a width of 8 bits.

23. (Original) The computer-readable medium of claim 14 wherein the catenated result has a width of 128 bits.

24. (Original) The computer-readable medium of claim 14 wherein for each field of the data selection operand, a relative location of the field within the data selection operand corresponds to a relative location of the predetermined position within the catenated result.

25. (Previously Presented) The computer-readable medium of claim 14 wherein at least some of the instructions further include a group floating point multiply instruction for multiplying floating point data in a programmable processor, the group floating point multiply instruction capable of instructing the computer to perform operations comprising:

decoding the group floating point multiply instruction specifying a third and a fourth register each containing a plurality of floating-point operands;

multiplying the plurality of floating point operands in the third register by the plurality of operands in the fourth register to produce a plurality of products; and

providing the plurality of products to partitioned fields of a result register as a catenated result.

26. (Currently Amended) A computer-readable medium:

having instructions that instruct a computer system to perform operations,

at least some of the instructions including a group element selection instruction for selectively arranging data in a programmable processor, the group element selection instruction capable of instructing a computer to perform operations comprising:

decoding the group element selection instruction specifying a data selection operand and a first register having a register width, the first register providing a plurality of data elements each having an elemental width smaller than the register width, the data selection operand comprising a plurality of fields each independently selecting one of the plurality of data elements; and

for each field of the data selection operand, providing the data element selected by the field to a predetermined position in a catenated result.

27 - 39. (Canceled)

40. (Currently Amended) A method of processing data in a programmable processor, the method comprising:

decoding a single instruction specifying a plurality of registers each having a register width, the plurality of registers storing a plurality of 8-bit data elements each having an elemental width smaller than the register width, an index register storing an index vector comprising a plurality of ~~equal-sized selectors~~ indices stored in partitioned fields of the index register and a destination register;

wherein each index in the index vector comprises a sufficient number of bits to represent a range of possible index values, the range of possible index values including a different index value for each of the plurality of data elements stored in the plurality of registers, allowing the index to select any data element from the plurality of data elements stored in the plurality of registers;

wherein each index in the index vector independently selects one of the data elements from the plurality of data elements stored in the plurality of registers; and

for each ~~selector~~ index in the index vector, providing a data element selected by the ~~selector~~ index to a predetermined position in the destination register.

41. (Previously Presented) The method set forth in claim 40 wherein the plurality of registers comprises two registers.

42. (Previously Presented) The method set forth in claim 40 wherein the plurality of registers comprises two 64-bit registers storing a combined total of sixteen 8-bit data elements.

43. (Currently Amended) The method set forth in claim 40 wherein the number of ~~selectors~~ indices stored in the index register is equal to the number of predetermined positions in the destination register.

44. (Previously Presented) The method set forth in claim 40 wherein the index register is a 64-bit register.

45. (Currently Amended) The method set forth in claim 40 wherein the index vector comprises n equal-sized ~~selectors~~ indices and the destination register comprises n equal-sized predetermined positions.

46. (Currently Amended) The method set forth in claim 45 wherein the ~~selector~~ index stored in a lowest order set of bits of the index register provides a data element to a lowest order set of bits of the destination register, the ~~selector~~ index in a second lowest order set of bits of the index register provide a data element to a second lowest order set of bits of the destination register and the ~~selector~~ index stored in a highest order set of bits of the index register provides a data element to a highest order set of bits of the destination register.

47. (Previously Presented) The method set forth in claim 40 wherein the destination register is a 128-bit register.



48. (Currently Amended) The method set forth in claim 40 wherein each of the equal-sized ~~selectors~~ indices stored in partitioned fields of the index register is a 4-bit ~~selector~~ index.

49. (Currently Amended) The method set forth in claim 40 wherein the index register stores sixteen 4-bit ~~selectors~~ indices.

50. (Currently Amended) A method of processing data in a programmable processor, the method comprising:

decoding a single instruction specifying a first register storing a first plurality of ~~8-bit~~ data elements, a second register storing a second plurality of ~~8-bit~~ data elements, an index register storing an index vector comprising a plurality of ~~equal-sized selectors~~ indices stored in partitioned fields of the index register and a destination register;

wherein each of the first and second registers has a register width, and each of the first and second plurality of data elements has an elemental width smaller than the register width;

wherein each index in the index vector comprises a sufficient number of bits to represent a range of possible index values, the range of possible index values including a different index value for each of the first and second pluralities of data elements stored in the first and second pluralities of registers, allowing the index to select any data element from the first and second pluralities of data elements stored in the first and second pluralities of registers;

wherein each index in the index vector independently selects one of the data elements from the first and second pluralities of data elements stored in the first and second pluralities of registers; and

for each ~~selector~~ index in the index vector, providing a data element from one of the first or second plurality of ~~8-bit~~ data elements selected by the ~~selector~~ index to a predetermined ~~8-bit~~ position in the destination register, wherein the predetermined positions are contiguous blocks of bits that take up an entire width of the destination register.

51. (Currently Amended) The method set forth in claim 50 wherein the first and second registers are 64-bit registers, the index register is a 64-bit register and each ~~selector~~ index stored in the index register has a sufficient number of bits to select any one of the 8-bit data elements in the first or second pluralities of 8-bit data elements.

52. (Previously Presented) The method set forth in claim 50 wherein the destination register is a 128-bit register.

53. (Currently Amended) The method set forth in claim 50 wherein each of the equal-sized ~~selectors~~ indices stored in partitioned fields of the index register is a 4-bit ~~selector~~ index.

54. (Currently Amended) A computer-readable medium having stored therein a plurality of instructions that cause a computer processor having registers to perform operations on data elements stored in registers within the processor, the plurality of instructions comprising:

an instruction specifying a plurality of registers each having a register width, the plurality of registers storing a plurality of 8-bit data elements each having an elemental width smaller than the register width, an index register storing an index vector comprising a plurality of ~~equal-sized~~ selectors indices stored in partitioned fields of the index register and a destination register;

wherein each index in the index vector comprises a sufficient number of bits to represent a range of possible index values, the range of possible index values including a different index value for each of the plurality of data elements stored in the plurality of registers, allowing the index to select any data element from the plurality of data elements stored in the plurality of registers;

wherein each index in the index vector independently selects one of the data elements from the plurality of data elements stored in the plurality of registers; and

wherein for each ~~selector~~ index in the index vector, the instruction causes the computer processor to provide a data element selected by the ~~selector~~ index to a predetermined position in the destination register.

55. (Previously Presented) The computer-readable medium set forth in claim 54 wherein the plurality of registers comprises two registers.

56. (Previously Presented) The computer-readable medium set forth in claim 54 wherein the plurality of registers comprises two 64-bit registers storing a combined total of sixteen 8-bit data elements.

57. (Currently Amended) The computer-readable medium set forth in claim 54 wherein the number of ~~selectors~~ indices stored in the index register is equal to the number of predetermined positions in the destination register.

58. (Previously Presented) The computer-readable medium set forth in claim 54 wherein the index register is a 64-bit register.

59. (Currently Amended) The computer-readable medium set forth in claim 54 wherein the index vector comprises n equal-sized ~~selectors~~ indices and the destination register comprises n equal-sized predetermined positions.

60. (Currently Amended) The computer-readable medium set forth in claim 59 wherein the ~~selector~~ index stored in a lowest order set of bits of the index register provides a data element to a lowest order set of bits of the destination register, the ~~selector~~ index in a second lowest order set of bits of the index register provide a data element to a second lowest order set of bits of the destination register and the ~~selector~~ index stored in a highest order set of bits of the index register provides a data element to a highest order set of bits of the destination register.

61. (Previously Presented) The computer-readable medium set forth in claim 54 wherein the destination register is a 128-bit register.

62. (Currently Amended) The computer-readable medium set forth in claim 54 wherein each of the equal-sized ~~selectors~~ indices stored in partitioned fields of the index register is a 4-bit ~~selector~~ index.

63. (Currently Amended) The computer-readable medium set forth in claim 54 wherein the index register stores sixteen 4-bit ~~selectors~~ indices.

64. (Currently Amended) A computer-readable medium having stored therein a plurality of instructions that cause a computer processor having registers to perform operations on data elements stored in registers within the processor, the plurality of instructions comprising:

an instruction specifying a first register storing a first plurality of 8-bit data elements, a second register storing a second plurality of 8-bit data elements, an index register storing an index vector comprising a plurality of ~~equal-sized selectors~~ indices stored in partitioned fields of the index register and a destination register;

wherein each of the first and second registers has a register width, and each of the first and second plurality of data elements has an elemental width smaller than the register width;

wherein each index in the index vector comprises a sufficient number of bits to represent a range of possible index values, the range of possible index values including a different index value for each of the first and second pluralities of data elements stored in the first and second pluralities of registers, allowing the index to select any data element from the first and second pluralities of data elements stored in the first and second pluralities of registers;

wherein each index in the index vector independently selects one of the data elements from the first and second pluralities of data elements stored in the first and second pluralities of registers; and

wherein for each ~~selector~~ index in the index vector, the instruction causes the computer processor to provide a data element from one of the first or second plurality of 8-bit data elements selected by the ~~selector~~ index to a predetermined 8-bit position in the destination register, wherein the predetermined positions are contiguous blocks of bits that take up an entire width of the destination register.

65. (Currently Amended) The computer-readable medium set forth in claim 64 wherein the first and second registers are 64-bit registers, the index register is a 64-bit register and each ~~selector~~ index stored in the index register has a sufficient number of bits to select any one of the 8-bit data elements in the first or second pluralities of 8-bit data elements.

66. (Previously Presented) The computer-readable medium set forth in claim 64 wherein the destination register is a 128-bit register.

67. (Currently Amended) The computer-readable medium set forth in claim 64 wherein each of the equal-sized ~~selectors~~ indices stored in partitioned fields of the index register is a 4-bit ~~selector~~ index.